Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 11462–11469

Automatic Multi-Robot Control Design and Optimization Leveraging Multi-Level Modeling: An Exploration Case Study ⋆

Wakana Endo *,**[1] **Cyrill Baumann** **[1] **Hajime Asama** *
Alcherio Martinoli **

* *Department of Precision Engineering, School of Engineering, The University of Tokyo, Japan, (e-mail: lastname@robot.t.u-tokyo.ac.jp).*
** *Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland, (e-mail: firstname.lastname@epfl.ch).*
[1] *These two authors contributed equally to this work.*

Abstract: In this work, we demonstrate the applicability of a recently proposed automatic synthesis approach for behavioral arbitrators based on Probabilistic Finite State Machines (PFSMs) for a multi-robot scenario. More specifically, a behavior-based controller for a multi-robot exploration scenario is automatically synthesized using a predefined set of basic behaviors and conditions. A key feature of the used synthesis approach is the tailored use of two modeling levels of the scenario, microscopic and submicroscopic, to significantly reduce the computational effort. Furthermore, the modeling is extended by a simplified macroscopic model of the scenario to analytically evaluate the best *achievable* performance given an ideal controller, taking into account real-world constraints such as limited speed and localization. Taking advantage of the interpretability of the synthesized PFSM-based arbitrators, individualistic and collaborative controllers are analyzed separately to provide insights into the theoretical and experimental effects of collaboration for the considered case study. The obtained results show that the PFSM-based synthesis approach is also suitable for multi-robot scenarios, and in particular that the collaborative solution can compete with a manually designed and fine-tuned solution.

Keywords: Multi-Level Modeling, Probabilistic Modeling, Learning for Control, Metaheuristic Optimization, Multi-Robot Systems, Behavior-Based Control

1. INTRODUCTION

Despite the ever-growing number of robots being used for increasingly complex tasks, many of these systems are still designed manually. Only recently, approaches for designing automatic control systems such as Birattari et al. (2021) or Mukhlish et al. (2018) have emerged, largely due to the increase of available computing power. Artificial Neural Networks (ANNs) combined with optimization strategies (e.g., reinforcement learning, evolutionary algorithms, metaheuristic optimizers) are currently the most popular approach for such automatically designed systems. However, ANNs have significant drawbacks, including difficulties in understanding, verifying, and analyzing their formal properties. Although progress has been made in this direction, for example by Borg et al. (2018), these limitations remain significant for critical applications. In this work, we thus focus on behavior-based controllers using behavioral arbitrators based on Probabilistic Finite State Machines (PFSMs) that are fully readable and verifiable.

In Francesca et al. (2014), Birattari et al. (2021) a framework has been introduced to automatically generate behavior-based controllers for robot swarms. Using a predefined set of basic behaviors and conditions, as well as a dedicated grammar, the synthesis problem is translated into an optimization problem. Ferrante et al. (2013) proposes a similar framework, but relying on evolutionary algorithms. Similarly, Neupane and Goodrich (2019) leverage grammatical evolution to obtain behavior tree controllers based on a similar set of basic behaviors and conditions. While the use of basic behaviors and conditions might be perceived as a limitation on the richness of the search space explorable by machine learning, such an approach offers further advantages in addition to the readability and verifiability mentioned above. For instance, a significant reduction of the reality gap, that is the difference between simulated and real-world experiments, can be achieved, due to the possibility of calibrating individual behaviors in simulation with respect to the reality.

The resulting behavior-based controllers are competitive in all the above works. However, their automatic generation involves a high computational cost as a result of the extensive use of high-fidelity simulations of the scenarios

⋆ This work is partially supported by Mitsubishi Electric Corporation, Japan, and partially by the University of Tokyo Go Global Scholarship.

10.1016/j.ifacol.2023.10.435

(submicroscopic level). A significant first progress towards the reduction of this computational cost has been made in Baumann et al. (2022a), where a more abstract model of the scenario (microscopic level) is leveraged to synthesize the structure of the PFSM-based arbitrator. The PFSM's parameters are then learned in a second step using high-fidelity simulation. Reductions of up to 75% in the computational cost were achieved, without degradation of the resulting control solution. However, the effectiveness of the approach was only demonstrated for a relatively simple single-robot scenario related to a foraging task.

In this work, we aim to demonstrate the viability of the automatic PFSM-synthesis framework, enhanced with the two-step modeling approach as introduced in Baumann et al. (2022a), for a significantly more complex scenario dealing with multiple robots. More concretely, this scenario involves a multi-robot system engaged in an exploration mission. The arena in which robots are deployed includes obstacles and several Points Of Interest (POI) that need to be discovered by the robots. Similar exploration scenarios have previously been addressed in the literature. In Maza and Ollero (2007), for example, the area is divided a priori, with each robot exploring its assigned region. However, this requires prior knowledge of the entire area and a centralized allocation process, which limits its applicability. Other works, often based on cooperative multi-robot techniques, do not require a map in advance. For example, Jatmiko et al. (2007) draw inspiration from the Particle Swarm Optimization algorithm to coordinate robots. In Dutta et al. (2019) and Hayat et al. (2017), the additional task of maintaining network connectivity was added, leading to a multi-objective problem.

In contrast to previous contributions to exploration scenarios, our work considers also robot failures. Robots fail with a given probability and can be repaired through interaction with other robots, similar to Christensen et al. (2009). This leaves the optimization algorithm the freedom to converge to an individualistic or collaborative behavior. Using additional modeling abstraction (macroscopic level), we analyze the scenario for both types of controllers to obtain lower bounds for the achievable cost. Together with a manually designed and fine-tuned solution, this serves as a comparison for the automatically optimized PFSM-based arbitrators.

The remainder of this paper is organized as follows. First, we give an overview of the modeling techniques and the synthesis framework used. Then, we introduce the considered scenario, the optimization problem, the available behavioral library, the dedicated microscopic model, and the experimental setup. Finally, we present and discuss the results obtained both in simulation and reality, before ending with some conclusive remarks. The development of the macroscopic model used for calculating the lower cost bounds can be found in Appendix A.

2. METHODS

In order to achieve its computational speedup, the PFSM-based arbitrator synthesis framework introduced in Baumann et al. (2022a) leverages both submicroscopic and microscopic modeling levels. They are part of the multi-level modeling framework for multi-robot systems, first introduced for robotic systems in Martinoli et al. (2004). The submicroscopic model (sub-μM) corresponds to a high-fidelity simulation of reality. The microscopic model (μM) also represents each robot individually, but only captures relevant robot features and includes numerous simplifications at the individual robot level. For example, individual sensor readings might get grouped into omnidirectional sensing information. Finally, at the macroscopic modeling level, the whole multi-robot system is represented as a single entity, usually leveraging a mean-field approach to obtain a mathematical representation.

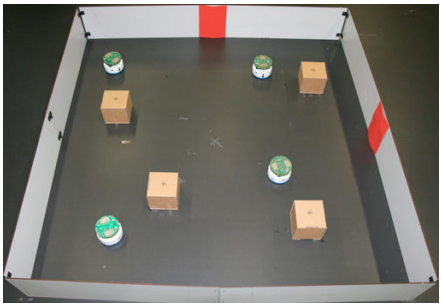

Fig. 1. Experimental arena for real robots.

The synthesis framework itself is based on a two-step approach, with each step leveraging one modeling level. First, the discrete parameters of the PFSM encoding its structure are optimized using a μM. Intuitively, this corresponds to choosing, from a library of available basic behaviors and conditions, the best suited combination of them for a given scenario. Second, the continuous parameters of the PFSM (i.e. probabilities) as well as specific behavioral parameters (e.g., sensor thresholds) are optimized using the more accurate sub-μM, corresponding to a fine-tuning of the behaviors for the task.

2.1 Scenario

We consider a scenario in which N_r wheeled robots are randomly placed in a square arena populated with obstacles, as illustrated in Figure 1. The robots are then tasked to find as many red POIs on the walls of the arena as possible and report their detection via wireless communication to a central hub. The robots are considered purely reactive, memoryless, operating in an area without access to Global Navigation Satellite Systems (GNSS), and endowed uniquely with local sensing capabilities and limited-range communication among themselves. The positions of the POIs do not change over time to keep the experimental setup simple. Furthermore, while successive reports of a given robot for the same POI are discarded, new nonsubsequent rediscoveries of the same POI are considered valuable. This effectively results in having the robots to constantly find the POIs anew using randomized navigation, as no deliberative planning is possible under these assumptions. To further increase the complexity of the scenario, we assume that the robots can randomly fail with probability P_f. Although wheeled robots can have various types of failure, we consider only a specific type here, namely one making robots unable to conduct any task and forcing them to stop. The failure can be "repaired" through an exchange of messages with another robot in proximity.

Table 1. Set of predefined behaviors.

Behavior	Symbol	Parameter 1	Parameter 2
Collision avoidance	CAB	agressivity	distance threshold
Stop	SPB	unused	unused
Rescue	RCB	speed	heading sensitivity
Failure	FLB	unused	unused
Straight	SRB	speed	unused
Random walk	RWB	speed	unused
Random turn	RTB	speed factor	interval
Report	RPB	unused	unused

2.2 Optimization problem

During the PFSM synthesis, our aim is to find an "optimal" behavioral arbitration solution θ^*, that minimizes a given cost function σ. That is, we look for

$$\theta^* = \arg\min_{\theta \in \Theta}(\sigma|\theta) \tag{1}$$

Where Θ corresponds to the set of all possible PFSMs, including structure and continuous parameters of both the PFSM and the underlying behaviors and conditions (*Parameters* in Tables 1 and 2), that can be constructed for a given number of PFSM states and a given behavior and condition library. Each PFSM is concretely encoded as a mixed-discrete vector, using the grammar introduced in Baumann et al. (2022a), with an additional parameter per transition for the transition probabilities. Note that the *Failure* behavior is not part of the optimization process, but is added to every θ prior to its evaluation, as we consider failure unavoidable independently of the PFSM.

For our multi-robot system, we define σ as function of three factors, all measurable for each individual robot i:

- the distance traveled (incorporated in σ^i_{mv});
- the number of state transitions (incorporated in σ^i_{tr});
- the number of POI reported (incorporated in σ^i_{rp}).

with σ being the average of the three factors:

$$\sigma = \frac{1}{N_r}\sum_{i=1}^{N_r}\left(\sigma^i_{rp} + \sigma^i_{tr} + \sigma^i_{mv}\right) \tag{2}$$

and the individual cost factors calculated as follows:

$$\sigma^i_{mv} = \frac{1}{N_k}\sum_{k=1}^{N_k}\left(1 - \frac{v[k]}{v_{max}}\right) \tag{3}$$

$$\sigma^i_{tr} = \begin{cases} 1 & \text{if } n^i_{tr} > N_k/5 \\ 0 & \text{if } n^i_{tr} \leq N_k/5 \end{cases} \tag{4}$$

$$\sigma^i_{rp} = \exp(-(n^i_{rp})/10) \tag{5}$$

where N_r is the number of robots; $v[k] = \frac{\left\|X^i[k] - X^i[k-1]\right\|}{\Delta T}$, with $X^i[k]$ the position of robot i at time step k and ΔT the duration of a time step, the robot speed computed in the sub-μM and real experiments, and $v[k] = \bar{v}$ the average speed of the robot while moving (and 0 otherwise), in the μM; N_k the number of time steps considered in the experiment; n^i_{tr} the sum of PFSM transitions recorded for robot i during the simulation, and n^i_{rp} the sum of reports received from robot i.

2.3 Behavior and condition libraries

The library of basic behaviors and conditions, available for the optimization algorithm to build a PFSM from, consists of eight distinct behaviors and eleven conditions. The behaviors are listed in Table 1. As most are self-explanatory, we omit their explanation here and focus on the more elaborate ones instead:

- The *Rescue Behavior* (RCB) consists of two internal procedures: approach and repair. Given a broken robot within its local communication range R_{com}, the rescuing robot approaches the broken robot to a distance R_{rsc} considered sufficient to carry out the rescue operation. The broken robot then informs the rescuing robot that its failure has been repaired. In case no broken robot is in range, the behavior is equivalent to the Stop Behavior (SPB).
- In the *Failure Behavior* (FLB), a robot stops its motion but continuously emits an emergency message through local communication.
- In the *Report Behavior* (RPB), a robot reports a found POI to a virtual central "supervisor" node, using global communication. As mentioned above, repeated reports about the same POI by the same robot are discarded until a new POI is found first.

Table 2. Set of predefined perceptual states determining most of the conditions.

Perceptual State	Symbol	Parameter
Default State	DSP	unused
Obstacle in range	OOP	sensor threshold
No obstacle in range	NOP	sensor threshold
Robot repaired	RPP	unused
Broken robot in range	BRP	unused
Repair finished	RFP	unused
POI in range	PRP	unused
No POI in range	NPP	unused
Failure	FFP	unused

The conditions are not listed explicitly, but most of them are straightforward to be constructed from the perceptual states in Table 2: for a given robot and perceptual state, the corresponding condition returns *true* if the robot is in the corresponding perceptual state. We denote this condition with ***C* (i.e. the condition corresponding to OOP is OOC). The more elaborate conditions are:

- The *Robot Repaired Condition* (RPC) becomes true when a robot is rescued by a surrounding robot and is able to transition to another state in its PFSM.
- The *Repair Finished Condition* (RFC) is fulfilled when a robot completes a rescue and gets the termination message from the former broken robot.
- In the *POI in Range Condition* (PRC) a robot checks if there is a POI in front using the on-board camera. When a predefined percentage (40% in our implementation) of the camera's field of view is occupied by the color of the POI, the robot is ready to report the POI found, and this condition becomes true.
- The *Always True Condition* (ATC) is not derived from any perceptual state but simply always returns true without a parameter.
- The *Timer Condition* (TTC) is not derived from any perceptual state either. It returns true after $1/parameter$ s.

As specified in Tables 1 and 2, each behavior and condition, directly coupled with a perceptual state, accepts up to two and one continuous parameters, respectively.

Table 3. Probability of transition conditions at each behavior.

Probability	Active Behavior							
	CAB	SPB	RCB	FLB	SRB	RWB	RTB	RPB
$P_{NOP-OOP}$	P_{OOP}	0	0	0	P_{OOP}	P_{OOP}	0	0
$P_{OOP-NOP}$	$1/T_{OOP}^{CAB}$	0	0	0	0	$1/T_{OOP}^{RWB}$	0	0
$P_{DSP-BRP}$	$N_f P_{BRP}$	0	0	0	$N_f P_{BRP}$	$N_f P_{BRP}$	0	0
$P_{BRP-RFP}$	0	0	T_{BRP}^{-1}	0	0	0	0	0
$P_{RFP-DSP}$	1	1	1	1	1	1	1	1
$P_{FFP-RPP}$	-	-	-	-	-	-	-	-
$P_{RPP-DSP}$	1	1	1	1	1	1	1	1
$P_{NPP-PRP}$	P_{PRP}	0	0	0	P_{PRP}	P_{PRP}	0	0
$P_{PRP-NPP}$	T_{report}^{-1}	1	1	1	T_{report}^{-1}	T_{report}^{-1}	T_{report}^{-1}	1

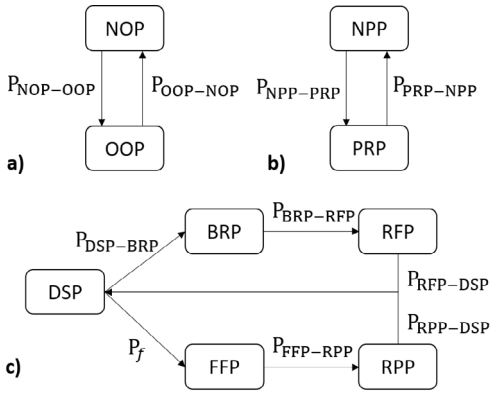

Fig. 2. PFSM of the perceptual situations used in the μM.

2.4 Microscopic model

Following the probabilistic framework of Martinoli et al. (2004), we can establish an agent-based probabilistic μM. However, differently from the previous work, the underlying robot controller structure is not available to build the structure of the model, as we are learning it. Therefore, similarly to Baumann et al. (2022a), we will leverage the perceptual situation of the robot, as illustrated in Figure 2, to define a specific state, where the probability of transition from one perceptual state to another depends on the currently *active* behavior.

Consistent with Correll and Martinoli (2004), when the well-mixedness assumption is satisfied (i.e., homogeneous spatial distribution of interactions among robots and between robots and the environment), the corresponding transition probabilities are calculated using geometric reasoning:

$$P_{OOP} = \frac{\bar{v} W_{ir}}{A_{rbt}} \frac{N_{obs} A_{obs} + A_{wal} + (N_{rbt} - 1) A_{rbt}}{A_a} \Delta T_{\mu M} \tag{6}$$

$$P_{BRP} = \frac{\bar{v} W_{com}}{A_{com}} \frac{N_f A_{com}}{A_a} \Delta T_{\mu M} = \frac{\bar{v} W_{com} N_f}{A_a} \Delta T_{\mu M} \tag{7}$$

$$P_{PRP} = \frac{\bar{v} W_{cam}}{A_{cam}} \frac{N_{poi} A_{cam}}{A_a} \frac{\theta_{poi}}{360} \Delta T_{\mu M} = \frac{\bar{v} N_{poi} W_{cam}}{A_a} \frac{\theta_{poi}}{360} \Delta T_{\mu M} \tag{8}$$

with $\Delta T_{\mu M}$ the time step duration in μM, N_{poi}, N_{obs} and N_{rbt} the number of POIs, obstacles and robots respectively, N_f the number of robots currently in FFP, $A_{wal}, A_{obs}, A_{rbt}$ the detection areas of the corresponding objects (walls, obstacles and robots, respectively), and A_a the entire area of the arena. In the same way, we define the different detection widths W of the corresponding sensors in the scenario. W_{ir} corresponds to the infrared proximity sensors, used to detect obstacles and walls, W_{com} to the radio communication, leveraged for detecting broken robots, and W_{cam} defines the detection width of the camera, used to detect POIs. Given that, in contrast to the other sensing modalities, the camera can not be considered unidirectional, we further have θ_{poi} as the (horizontal) angle of view of the camera.

Table 4. Numerical parameters of the μM.

Symbol	Value	Symbol	Value
W_{ir}	0.46 m	$N_k^{\mu M}$	1′200
W_{cam}	0.42 m	$N_k^{sub-\mu M}$	3′750
W_{com}	1.2 m	N_k^{RR}	300
A_a	4 m^2	θ_{poi}	131°
A_{obs}	0.32 m^2	T_{OOP}^{CAB}	4.6 s
A_{poi}	0.14 m^2	T_{OOP}^{RWB}	13.7 s
A_{com}	1.1 m^2	T_{report}	0.76 s
A_{wal}	1.4 m^2	N_{obs}	4
A_{rbt}	0.17 m^2	N_r	4
$\Delta T_{sub-\mu M}$	0.032 s	N_{poi}	3
$\Delta T_{\mu M}$	0.10 s	v_{max}	0.24 m/s
ΔT_{RR}	0.40 s	$\bar{v}$	0.08 m/s
R_{com}	0.6 m	P_f	0.01
R_{rsc}	0.2 m		

When the well-mixedness assumption is only partially fulfilled, we leverage the sub-μM to estimate the concerned parameter. For instance, we measure an average event duration in the sub-μM and assume a Poisson distribution to estimate its probability of occurrence:

$$P_{OOP-NOP}^{CAB} = \frac{1}{T_{OOP}^{CAB}} \tag{9}$$

with T_{OOC}^{CAB} the recorded average time when in CAB to transition from OOP to NOP (see Table 3, row 2, column 1). Similarly, T_{report} denotes the average time a POI remains in range.

As the robot must first approach a broken robot to rescue it, the probability in which the rescue procedure ends is expressed by using the average approaching time $T_{BRP} = (R_{com} - R_{rsc})/\bar{v}$, with $\bar{v}$ the average robot speed and R_{com}, R_{rsc} the communication and rescuing range, respectively. As the duration of the rescue operation itself is negligible in this work, this probability can be expressed as T_{BRP}^{-1}.

Thanks to the reasoning above, all probabilities in Table 3 can be computed using the numerical values provided in Table 4. Note that Table 3 can be considered a lookup table to determine the state transition probabilities for the perceptual PFSM in Figure 2, as function of the currently active behavior. As the change from FFP to RPP is fully dependent of the actions of the *other* robots, $P_{FFP-RPP}$ does not have any probability. In fact, this transaction happens with probability N_f^{-1} whenever the BRP to RFP transition happens in *another* robot.

It is further worth noting that, while a μM could also be used to optimize some continuous behavior/condition parameters (e.g., speeds), others (e.g., sensor thresholds) cannot be captured without substantially increasing the model complexity. We therefore chose to only explicitly

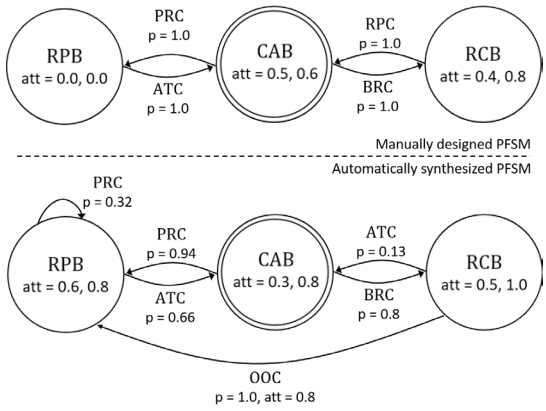

Fig. 3. The manually designed (top) and one of the twenty synthesized PFSMs (bottom). Double circles indicate the initial state. Only relevant (i.e. not *unused* in Tables 1 & 2) condition attributes are reported.

expose (and thus optimize) the discrete parameters of the PFSM arbitrator in the μM.

3. EXPERIMENTAL SETUP

To validate the applicability of the automatic synthesis framework of Baumann et al. (2022a) to our multi-robot system, we follow the two-step process as already outlined in Section 2: we first conduct the optimization of the PFSM structure using the μM, before optimizing the PFSM parameters using the sub-μM. We use Webots, Michel (2004), an open-source, high-fidelity simulator as implementation of the sub-μM. Note that we calibrated both sensors and actuators in Webots to match as closely as possible those of the real robots. Finally, using simulated and real experiments, we validate and compare the performances of the automatically optimized controller with the bounds obtained in Appendix A and a handcrafted and fine-tuned solution shown in Figure 3 (top).

We set the number of states in the PFSM controller to three, as our system analysis showed that an ideal collaborative controller needs at least three behaviors: RCB, RPB, and CAB. While PFSMs with more states are possible and might result in similar performance, an increased number of states would exponentially increase the computational cost required for synthesizing the solution.

We use a Mixed-Discrete Particle Swarm Optimization with Optimal Computing Budget Allocation (MDPSO-OCBA), Baumann and Martinoli (2022b), to optimize the PFSM-based arbitrator. The relevant meta-parameters for the metaheuristic optimization algorithm are reported in Table 5. We perform a total of twenty optimization runs with a budget of 50'000 and 2'000 evaluations of candidate solutions in the μM and the sub-μM, respectively. The resulting PFSMs are then re-evaluated ten times in the sub-μM. Leveraging the human readability of the resulting PFSMs, we then split the PFSMs into individualistic and collaborative solutions according to their structure. The five best performing candidate solutions for both categories are further re-evaluated five times in reality. These synthesized controllers are compared to a manually fine-tuned PFSM, which is evaluated ten times in the sub-μM and five times in reality. Additionally, using expert

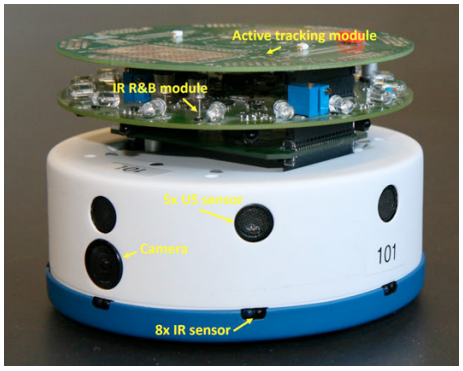

Fig. 4. Khepera IV robot with a custom range-and-bearing module as well as a tracking module. In this work, the US sensors are not used.

Table 5. Parameters for the MDPSO-OCBA optimization.

Description	step 1 (μM)	step 2 (sub-μM)
Number of particles	150	36
Maximum evaluation budget	50'000	2'000
Number of variables	15	18

knowledge of the underlying scenario, we build a simplified macroscopic model to provide a lower bound on expected performance ($\hat{\sigma}$) for an ideal individualistic ($\theta \in \Theta^*_{ind}$), respectively collaborative controller ($\theta \in \Theta^*_{col}$). The mathematical development of the macroscopic model can be found in Appendix A.

All experiments were conducted using either simulated or real Khepera IV robots. Khepera IV robots are differentially driven vehicles with a diameter of 14 cm and a maximum speed of ~ 81 cm/s. They are further equipped with eight infrared proximity sensors with a range of 20 cm that are used for obstacle avoidance, as well as a camera used for POI detection. As illustrated in Figure 4, the robots are enhanced with a custom range-and-bearing module, Pugh et al. (2009), for relative localization and an active marker module featuring two LEDs that allows for accurate tracking through the SwisTrack software, Lochmatter et al. (2008). Communication between the robots is leveraging UDP/IP, assumed to be error-free but limited to a range of R_{com} and is only used for identifying broken robots.

For both simulated and real robot experiments, we randomly place four Khepera IV robots in a 2 m $\times$ 2 m arena populated with four obstacles and three POI, as illustrated in Figure 1. The robots are then free to roam in the arena for 120 s, after which the three metrics associated to the cost function of Eq. 2 are calculated.

4. RESULTS AND DISCUSSION

Among the twenty optimized solutions, we found eight resulting individualistic PFSMs and twelve collaborative ones. As illustrated in Figure 3, the learned collaborative PFSM strongly ressemble the manual one. Figure 5 shows the learning rates of the PFSM arbitrators for both categories in comparison to the average cost of the manual solution in the sub-μM, as well as the two lower bounds obtained analytically. On the horizontal axis, we report equivalent sub-μM candidate evaluations, since an evaluation time using the μM is significantly shorter than the sub-μM one. Consistent with previous work, we observe

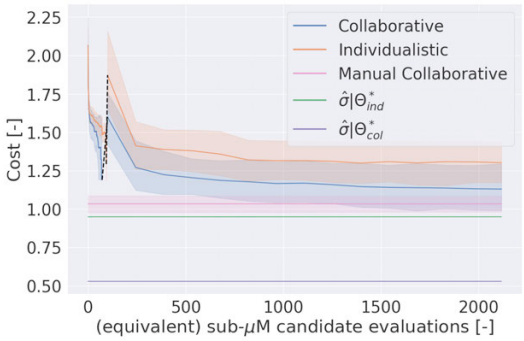

Fig. 5. Average and 95% confidence interval (shaded areas) of the global best fitness during MDPSO-OCBA learning among the solutions. The dotted line corresponds to the change from the first to the second optimization step.

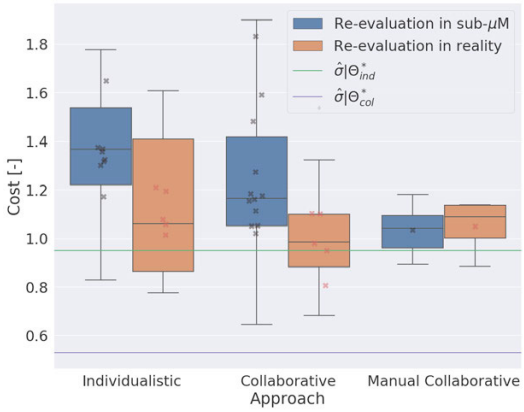

Fig. 6. Comparison of the re-evaluated PFSM-based arbitrators in both sub-μM and in reality. The crosses are the average cost of ten evaluations of each solution.

that the learning process in μM takes only a fraction of the total optimization budget, despite the 25 times larger evaluation budget as shown in Table 5, confirming the effectiveness of the two-step approach for this scenario. We further notice that collaborative solutions result in consistently lower cost than individualistic ones, meaning that they are more competitive and, sporadically, even outperforming the manual solution. We also note that there is a cost jump between the μM learning curve and the sub-μM learning curve. This is supposedly due to the significant influence of the PFSM's continuous parameters on the system performance.

Figure 6 shows the comparison of the re-evaluated PFSM controllers in both simulation and reality. Note that the re-evaluation in sub-μM was performed with all twenty solutions, whereas only the five best solutions per category were evaluated in real robot experiments [1]. We observe that several collaborative solutions achieved comparable performances to the manual solution. However, the average performance of the synthesized controllers is worse than the manual one. This is due to the complexity and stochasticity of the scenario which makes finding optimal continuous parameters challenging. We also conducted a Kruskal-Wallis Test which resulted in significant differences ($p < .01$) among the performance in the sub-μM simulation of the three approaches. Pairwise comparisons using Dunn's test indicated that the differences in performances between all pairs is statistically significant ($p < .05$). For real robots, the collaborative PFSMs seem to be slightly better than individualistic and manually designed ones. However, a Kruskal-Wallis test was conducted and revealed no significant differences ($p > .50$) among the three categories. This can be explained by the fact that we only evaluated the five best synthesized controllers per category in reality.

We further observe that some individualistic candidates resulted in a better cost than the analytically derived lower bound (c.f. Appendix A). This is because the lower bounds correspond to the expected *average* performances for an achievable optimal controller. Therefore, individual experimental runs may occasionally result in better performance, given the stochastic nature of the experiments (i.e. no failure occurs due to the robots being lucky, etc.). For the collaborative approach, however, we observe that the optimal value is never reached by any experimental run. This is supposedly due to a number of small assumptions in our macroscopic model (e.g., no obstacles in between a broken robot and a rescuer, being able to encounter and report POIs while rescuing, etc). While a more accurate model is possible, its complexity would substantially increase. Given that in this work, the purpose of the macroscopic model was to provide a lower bound to the achievable cost, we believe this additional faithfulness of the macroscopic model is not needed.

While both the performance of the synthesized PFSMs, as well as the synthesizing speed of this approach look promising, there are some challenges to it. Firstly, the approach requires a microscopic model of the scenario and the behavior and condition libraries. While, in theory, these latter could be reused, this is something we aim to demonstrate concretely in a future work. Secondly, it is worth noting that the creation of a microscopic model currently needs a substantial amount of expert knowledge. Consequently, a simplification or even automatisation of the modeling process would greatly improve the impact of the approach.

5. CONCLUSION

In this paper, we have demonstrated that a robust metaheuristic optimization algorithm can be coupled with multi-level modeling to achieve computationally efficient design and optimization of PFSM-based behavioral arbitrators for a complex multi-robot scenario. Taking advantage of the nature of the scenario and the interpretability of the synthesized PFSM-based arbitrators, individualistic and collaborative controllers could be identified and analyzed individually. We used a macroscopic model to obtain analytically the best achievable performances, which, together with a manually designed and fine-tuned controller, served as challenging comparison for the learned behavioral arbitrators. While differences in performance were observed in simulation, statistical tests revealed no statistically significant differences in performance between the manually designed and automatically synthesized PFSM-based behavioral arbitrators in real robot experiments.

[1] A video of real multi-robot experiments can be found on our research page: www.epfl.ch/labs/disal/research/mrsmodelingcontrol

REFERENCES

Baumann, C., Birch, H., and Martinoli, A. (2022a). Leveraging Multi-Level Modelling to Automatically Design Behavioral Arbitrators in Robotic Controllers. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 9318–9325.

Baumann, C. and Martinoli, A. (2022b). A Noise-Resistant Mixed-Discrete Particle Swarm Optimization Algorithm for the Automatic Design of Behavioral Arbitrators in Robotic Controllers. In *IEEE Congress on Evolutionary Computation*, 9 pages.

Birattari, M., Ligot, A., and Francesca, G. (2021). AutoMoDe: A Modular Approach to the Automatic Off-Line Design and Fine-Tuning of Control Software for Robot Swarms. In *Automated Design of Machine Learning and Search Algorithms*, Natural Computing Series, 73–90.

Borg, M., Englund, C., Wnuk, K., Duran, B., Levandowski, C., Gao, S., Tan, Y., Kaijser, H., Lönn, H., and Törnqvist, J. (2018). Safely Entering the Deep: A Review of Verification and Validation for Machine Learning and a Challenge Elicitation in the Automotive Industry. *arXiv:1812.05389*.

Christensen, A.L., OGrady, R., and Dorigo, M. (2009). From fireflies to fault-tolerant swarms of robots. *IEEE Transactions on Evolutionary Computation*, 13(4), 754–766.

Correll, N. and Martinoli, A. (2004). Collective Inspection of Regular Structures using a Swarm of Miniature Robots. In *Ninth International Symposium on Experimental Robotics*. Springer Tracts in Advanced Robotics (2006), pp. 375-385.

Dutta, A., Ghosh, A., and Kreidl, O.P. (2019). Multi-robot Informative Path Planning with Continuous Connectivity Constraints. In *2019 International Conference on Robotics and Automation*, 3245–3251.

Ferrante, E., Duenez-Guzman, E., Turgut, A., and Wenseleers, T. (2013). GESwarm: grammatical evolution for the automatic synthesis of collective behaviors in swarm robotics. In *ACM Genetic and Evolutionary Computation Conference*, 17–24.

Francesca, G., Brambilla, M., Brutschy, A., Trianni, V., and Birattari, M. (2014). AutoMoDe: a novel approach to the automatic design of control software for robot swarms. *Swarm Intelligence*, 8, 89–112.

Hayat, S., Yanmaz, E., Brown, T.X., and Bettstetter, C. (2017). Multi-objective UAV path planning for search and rescue. In *IEEE International Conference on Robotics and Automation*, 5569–5574.

Jatmiko, W., Sekiyama, K., and Fukuda, T. (2007). A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement. *IEEE Computational Intelligence Magazine*, 2(2), 37–51.

Lochmatter, T., Roduit, P., Cianci, C., Correll, N., Jacot, J., and Martinoli, A. (2008). SwisTrack - a flexible open source tracking software for multi-agent systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 4004–4010.

Martinoli, A., Easton, K., and Agassounon, W. (2004). Modeling Swarm Robotic Systems: a Case Study in Collaborative Distributed Manipulation. *The International Journal of Robotics Research*, 23(4-5), 415–436.

Maza, I. and Ollero, A. (2007). Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In *Distributed Autonomous Robotic Systems 6*, 221–230.

Michel, O. (2004). WebotsTM: professional mobile robot simulation. *International Journal of Advanced Robotic Systems*, 1, 39–42.

Mukhlish, F., Page, J., and Bain, M. (2018). Evolutionary-learning framework: improving automatic swarm robotics design. *International Journal of Intelligent Unmanned Systems*, 6(4), 197–215.

Neupane, A. and Goodrich, M.A. (2019). Learning swarm behaviors using grammatical evolution and behavior trees. In *International Joint Conference on Artificial Intelligence*, 513–520.

Pugh, J., Raemy, X., Favre, C., Falconi, R., and Martinoli, A. (2009). A Fast Onboard Relative Positioning Module for Multirobot Systems. *IEEE/ASME Transactions on Mechatronics*, 14(2), 151–162.

Appendix A. SYSTEM ANALYSIS USING A MACROSCOPIC MODEL

While the cost of a *perfect* controller is $\sigma = 0$, such a controller is not feasible in reality and therefore irrelevant for most practical applications. Consequently, we study here the best *achievable* performance by an ideal controller, subject to all real-world constraints of both the robot and the scenario, such as limited speed and no knowledge of the location of POIs. Naturally, such an ideal controller is highly specific to a particular mission and might not fully be achievable through a combination of the available behaviors listed in Table 1. However, such an idealized controller allows us to compute lower bounds. Given the idealized conditions and possible formal considerations, we perform this analysis using a macroscopic model of the multi-robot system, which is closely rooted to the μM presented above, and again follows Martinoli et al. (2004).

Formally, we are looking for $\Theta^* \subset \Theta$, the subset of controllers θ that results in optimal performance with respect to Eq.1. We further assume that the maximum speed used in the calculation of the cost function corresponds to the maximum speed that any controller can achieve.

We note that an ideal controller minimizes each of the three σ individually. That is:

$$\min(\sigma|\theta) \geq \min(\frac{1}{N_r}\sum_{i=1}^{N_r}\sigma_{rp}^i|\theta) + \min(\frac{1}{N_r}\sum_{i=1}^{N_r}\sigma_{tr}^i|\theta) + \min(\frac{1}{N_r N_k}\sum_{i=1}^{N_r}\sum_{k_1}^{N_k}\sigma_{mv}^i[k]|\theta)\ \forall\theta\in\Theta \quad \text{(A.1)}$$

Consequently, we will treat σ_{rp}, σ_{tr} and σ_{mv} independently for the remainder of this analysis. Given that we only consider homogeneous robots, we can further drop the superscript i. The expected cost function ($\hat{\sigma}$) can thus be expressed as $\hat{\sigma} = \hat{\sigma}_{mv} + \hat{\sigma}_{rp} + \hat{\sigma}_{tr}$.

Idealistic case In a first step, we assume that the robots do not collide or fail.

Lemma 1. *Θ^*_{ideal}, the set of optimal idealistic controllers is given by $\{\theta : \bar{v} = v_{max}\}$, with expected cost $\hat{\sigma}_{mv}|\Theta^*_{ideal} =$*

0, $\hat{\sigma}_{rp}|\Theta^*_{ideal} = \exp\left(-(N_r \frac{\theta_{poi} N_{poi} \bar{v} W^{cam} A_{poi} N_k}{360 A_{poi} A_a})/10\right)$ *and* $\hat{\sigma}_{tr}|\Theta^*_{ideal} = 0$.

Proof. Using Eq. 3, the set of ideal and achievable controllers with respect to σ_{mv}, is given by $\Theta^*_{ideal} = \Theta \cup \{\theta : \bar{v} = \arg\min_v(||\bar{v} - v_{max}||)\}$. Given the assumption of no knowledge about the location of POIs, the expected probability of encountering a POI can be defined geometrically using Eq. 8. The expected number of reports is thus $\hat{N}_{rp} = N_r N_k P_{PRP} = \bar{v}C$, with $C = \frac{N_r N_k \theta_{poi} W_{cam} N_{poi} A_{poi}}{360 A poi A_a}$ $\forall\theta$. Therefore, $\Theta^*_{ideal} \subset \arg\min_\theta(\sigma_{rp}|\theta) = \arg\min_\theta(e^{-\bar{v}C/10}|\theta)$. Furthermore, Θ^*_{ideal} needs to report the finding of a POI immediately and without delay, since σ_{rp} takes into account *reported* POI, but $\hat{N}_{rp}$ corresponds (so far) to POI found. We denote by Θ^*_{report} the set of controllers that meet this condition. Thus, $\Theta^*_{ideal} \subset \Theta^*_{report}$.

Given that $\bar{v} \geq 0$ $\forall\theta \in \Theta$, for both θ_{mv} and θ_{rp} we thus obtain $\Theta^*_{ideal} \subset \arg\max_\theta(\bar{v}|\theta)$. For this idealistic case, this further simplifies to $\Theta^*_{ideal} = \{\theta : \bar{v} = v_{max}\}$ and consequently to $(\hat{\sigma}_{mv}|\Theta^*_{ideal}) = 0$.

It is trivial to see that $(\hat{\sigma}_{tr}|\Theta^*_{rp}) = 0$ as long as a maximum of one state transition occurs every five time steps, which we assume to be easily satisfied by any Θ^*_{ideal}. □

Collision avoidance (CA) In a second step, we consider that robots collide. For the sake of simplicity, we assume that a robot, once in a collision, cannot move again, but that $\forall\theta \in \Theta^*_{ca}$, the set of behaviors including optimal CAB, robots do not enter into collisions.

Lemma 2. *For our case study, there exists $\Theta^*_{ca*} \subset \Theta^*_{ca}$, a subset of behaviors including optimal CAB, whose impact on σ is negligible.*

Proof. CAB is only necessary whenever a robot encounters either an obstacle, wall or another robot. The average number of time steps between such encounters is $\frac{1}{P_{OOP}}$, with P_{OOP} given by Eq. 6. Thus, the impact of the CAB is negligible if $N_{CA} \ll 1/P_{OOP}$, with N_{CA} the average number of time steps in which the CAB is active. □

Failure Given that robots fail with probability P_f, Lemma 1 has to be revisited, taking into account that $\bar{v} = 0$ for any robot in $Failure$ state. To leave the $Failure$ state, another robot in the *Rescue* behavior is necessary. We can distinguish between two types of optimal controllers: *individualistic* ones, which are not rescuing other robots, and *collaborative* ones which do rescues. They are defined as $\Theta^*_{ind} \subset \Theta^* \setminus \Theta_{rescue}$ and $\Theta^*_{col} \subset \Theta^* \cup \Theta_{rescue}$, respectively. In order to pursue our analysis, we introduce a simplified macroscopic model keeping track only of the average numbers of robots in *optimal movement*, $failure$ and *rescuing* state respectively:

$$N_r = N_{opt}[k] + N_{fal}[k] + N_{rsc}[k] \quad (A.2)$$

$$N_{opt}[k+1] = N_{opt}[k] + 2\Delta_{fix}[k] - \Delta_{opt-fal}[k] - \Delta_{rsc}[k] \quad (A.3)$$

$$N_{fal}[k+1] = N_{fal}[k] + \Delta_{fal}[k] - \Delta_{fix}[k] \quad (A.4)$$

$$N_{rsc}[k+1] = N_{rsc}[k] + \Delta_{rsc}[k] - \Delta_{fix}[k] - \Delta_{rsc-fal}[k] \quad (A.5)$$

The corresponding transitions per time step (the Δ's) can be computed leveraging the definitions introduced for the μM, which is coherent with Martinoli et al. (2004):

$$\Delta_{opt-fal}[k] = P_f N_{opt}[k] \quad (A.6)$$

$$\Delta_{rsc-fal}[k] = P_f N_{rsc}[k] \quad (A.7)$$

$$\Delta_{fal}[k] = \Delta_{opt-fal}[k] + \Delta_{rsc-fal}[k] \quad (A.8)$$

$$\Delta_{rsc}[k] = P_{BRP} N_{opt}[k] N_{fal}[k] \quad (A.9)$$

$$\Delta_{fix}[k] = \Delta_{rsc}[k - T_{BRP}](1 - P_f)^{T_{BRP}} \quad (A.10)$$

It is worth stressing, that the purpose of this macroscopic model is purely to analyze the expected behavior. Consequently, its explicitly exposed parameters are those anchored to the physical quantities characterizing the set-up and not the control design.

Lemma 3. *The expected costs of an optimal individualistic and collaborative controller depend only on the average speed $\bar{v}_{ind}, \bar{v}_{col}$ achieved through Θ^*_{ind} and Θ^*_{col}, respectively. The corresponding average speeds are:*

$$\bar{v}_{ind} = \frac{1 - (1 - P_f)^{N_k}}{N_k P_f} v_{max} \quad (A.11)$$

$$\bar{v}_{col} = \frac{N_{opt} + N_{rsc}}{N_r} v_{max} \quad (A.12)$$

Proof. Lemma 1 states that $\bar{v}$ should be maximized to minimize all cost functions. As P_f is uncorrelated with $\bar{v}$, this remains valid for the case including failures. Therefore, to find the performance of an optimal controller, it is sufficient to find the average speed $\bar{v}_{ind,col}$ achievable through Θ^*_{ind} or Θ^*_{col}, respectively. Consequently, we obtain $\bar{v} = \bar{v}_{ind,col}$ instead of v_{max} in Lemma 1.

For any given robot, the probability that no failure occurs up to the time step k is $(1 - P_f)^{k-1}$. Therefore, the expected average speed of an individualistic controller can be defined as

$$\bar{v}|\theta \in \Theta^*_{ind} = \frac{1}{N_k} \sum_{k=1}^{N_k} \left(v_{max}(1 - P_f)^{k-1}\right) \quad (A.13)$$

According to the characteristic of the geometric series, this can be simplified to

$$\bar{v}|\theta \in \Theta^*_{ind} = \frac{v_{max}(1 - (1 - P_f)^{N_k})}{N_k P_f} \quad (A.14)$$

For the collaborative case, the situation is more complex due to the rescue of failed robots. However, it is reasonable to assume that $\Theta_{rescue} \cup \{\theta : \bar{v} = v_{max}\} \neq \emptyset$. Therefore, $\bar{v}_{col} = \frac{N_{opt} + N_{rsc}}{N_r} v_{max}$. With N_{opt} and N_{rsc} found using steady-state analysis of Eqs. A.2 - A.5:

$$2\Delta_{fix} = \Delta_{opt-fal} + \Delta_{rsc} \quad (A.15)$$

$$\Delta_{fal} = \Delta_{fix} \quad (A.16)$$

$$\Delta_{rsc} = \Delta_{fix} + \Delta_{rsc-fal} \quad (A.17)$$

and thus, using Eqs. A.6 - A.10, we obtain:

$$N_{opt} = \frac{(2N_r P_{BRP}(1 - P_f)^{T_{BRP}} - P_f - N P_{BRP}}{P_{BRP}(1 - P_f)^{T_{BRP}}} \quad (A.18)$$

$$N_{rsc} = \frac{((1 - P_f)^{T_{BRP}} - 1)(P_f + N_r P_g(1 - 2(1 - P_f)^{T_{BRP}}))}{(1 - P_f)^{T_{BRP}} P_{BRP}(2(1 - P_f)^{T_{BRP}} - 1)}$$

$$N_{fal} = \frac{P_f}{P_{BRP}(2(1 - P_f)^{T_{BRP}} - 1)} \quad (A.19)$$

□